Alexander Garcia June 28, 2017

Assignment-5

1.
$$I = \int_{-1}^{1} e^{-2x} dx$$

(a) Approximation using the composite trapezoidal rule.
The general formula for the composite trapezoidal rule is

$$I(f) = \frac{h}{2}[f(a) + f(h) + 2\sum_{j=1}^{k-1} f(a + jp)]$$

where h is the width of the total interval, p is the node width, and k is the total number of nodes. In this case, $h = p = 0.5$, $k = 4$, and $f = e^{-2x}$.
Then, according to the rule,

$$I(f) = \frac{0.5}{2}[f(-1) + f(1) + 2\sum_{j=1}^{3} f(a + jp)]$$

or

$$I(f) = \frac{1}{4}[e^{-1} + e + 2[e^{-1+0.5} + e^{-1+1} + e^{-1+1.5}]]$$

When the expression is evaluated, we get the composite trapezoidal approximation of the integral to be
$$I(f) \approx 2.399166\ldots$$

We know that the error for this formula can be expressed by $-\frac{h^2}{12}(b-a)f''(\eta)$, where η is an unknown location within the interval. Rather than use an unknown location, we can bound the error by the maximum of the second derivative over the interval.

$$f''(x) = 4e^{-2x} \leq 4e^2 \quad x \in [-1, 1]$$

Therefore, the maximum error of this approximation is

$$\frac{0.5^2}{12}(1 - (-1))(4e^2) \approx 1.231509\ldots$$

which makes our result not very significant.

(b) Approximation using composite Simpson's rule

We can define composite Simpson's rule in this case as the following

$$I(f) = \frac{h}{3}\sum_{j=2}^{8}(f(a+(j-2)h) + 4f(a+(j-1)h) + f(a+jh))$$

In our case, $k = 4$, and $p = \frac{b-a}{k}$, so $h = \frac{p}{2}$. We can write out the full summation then as

$$\frac{h}{3}[\begin{array}{ll} f(-1) + 4f(-0.75) + f(-0.5) & + \\ f(-0.5) + 4f(-0.25) + f(0) & + \\ f(0) + 4f(0.25) + f(0.5) & + \\ f(0.5) + 4f(0.75) + f(1) & \end{array}]$$

The overall error of Simpson's rule approximation can be written as

$$E \leq -\frac{b-a}{180}h^4 max(f''(x)) \quad x \in [-1, 1]$$

We can easily see that the $max(f''(x)) = 16e^2$ for our interval, so the upper bound on the error is

We can then calculate the upper bound on the error.

$$E \leq -\frac{2}{180}(0.25^4)(16e^2)$$
$$E \leq -0.00513129\ldots$$

This makes composite Simpson's approximation a much better estimate of the integral than composite trapezoidal approximation.

(c) For the composite trapezoidal rule, we are given that the error is defined by

$$|E| \leq \frac{h^2}{12}(b-a)max(f''(x)) < 10^{-6} \quad x \in [a, b]$$

As we are simply solving for the node spacing h, we can rearrange this equation to gain an expression for its value.

$$h < \sqrt{\frac{12\delta}{(b-a)M}}$$

where

$M = max(f''(x))\ x \in [-1, 1] = 4e^2$

$\delta = 10^{-6}$, and

$b - a = 2$.

When these values quantities are used, we get

$$h < (\sqrt{\frac{12 * 10^{-6}}{2 * 4e^2}} \approx 0.000450558\ldots)$$

(d) We have a similar expression for the error of Simpson's rule

$$|E| \leq \frac{b-a}{180} h^4 max(f^{(iv)}(x)) < 10^{-6} \quad x \in [a,b]$$

We can again rearrange this equation to solve for h, the minimum node spacing to ensure an error of $\leq 10^{-6}$

$$h < (\frac{180\delta}{(b-a)M})^{1/4}$$

where $M = max(f^{(iv)}(x)) \quad x \in [-1,1] = 16e^2$
$\delta = 10^{-6}$, and
$b - a = 2$.
This inequality gives us a minimum node spacing of

$$h < ((\frac{180 * 10^{-6}}{2 * 16e^2})^{1/4} \approx 0.0295381\ldots)$$

2. Holmes 6.18

(a) $I_S(n) = \frac{2}{3} I_T(n) + \frac{1}{3} I_M(n/2)$
The approximation of this integral must take place over an interval $[a,b]$, the size of which determines the node spacing h. For n nodes, $h = \frac{b-a}{n}$, and for $n/2$ nodes, $h = 2\frac{b-a}{n}$. Going forward, we use these values of h for $I_T(n)$ and $I_M(n)$ respectively.

We know the definition of $I_M(n)$ as

$$I_M(n) = h[f(a+h/2) + f(a+h) + f(a+3h/2) + f(a+2h) + \ldots + f(a+(n-1)h/2)]$$

Considering the fact that $h = 2\frac{b-a}{n}$ when we take $I_M(n/2)$, we can rewrite this equation as

$$I_M(n/2) = 2\frac{b-a}{n}[f(a + 2\frac{b-a}{n}) + f(a + 4\frac{b-a}{n}) + \ldots]$$

and

$$\frac{1}{3} I_M(n/2) = \frac{2(b-a)}{3n}[f(a + 2\frac{b-a}{n}) + \ldots]$$

We also know the definition of $I_T(n)$

$$I_T(n) = \frac{b-a}{2n}[f(a) + f(b) + 2f(a + \frac{b-a}{n}) + 2f(a + 2\frac{b-a}{n}) + 2f(a + 3\frac{b-a}{n}) + \ldots]$$

and

$$\frac{2}{3} I_T(n) = \frac{b-a}{3n}[f(a) + f(b) + 2f(a + \frac{b-a}{n}) + 2f(a + 2\frac{b-a}{n}) + \ldots]$$

These two sequences share terms that are of the form $f(a + 2i\frac{b-a}{n})$, meaning the even terms are seen twice. In $I_T(n)$, there are also multiplied by a factor of 2, meaning that in total, each even term appears 4 times. The odd terms in $I_T(n)$, then, are only found in $I_T(n)$, but are still multiplied by 2, so each odd term appears 2 times. Each end point $f(a), f(b)$ only appears in $I_T(n)$, so each has a weight of 1. When the two sequences are combined, we get

$$\frac{(b-a)}{3n}[f(a) + f(b) + 4f(x_{even}) + 2f(x_{odd})]$$

where $f(x_{odd}) = f(a + odd\frac{b-a}{n})$ and $f(x_{even}) = a + even\frac{b-a}{n}$. This is the definition of composite Simpson's rule.

(b) $I_S(n) = \frac{4}{3}I_T(n) - \frac{1}{3}I_M(n/2)$

A very similar idea is used to demonstrate this alternate definition of composite Simpson's rule.

$$\frac{1}{3}I_M(n/2) = \frac{2(b-a)}{3n}[f(a + 2\frac{b-a}{n}) + \ldots]$$

and

$$\frac{4}{3}I_T(n) = \frac{2(b-a)}{3n}[f(a) + f(b) + 2f(a + \frac{b-a}{n}) + 2f(a + 2\frac{b-a}{n}) + \ldots]$$

Again, once the difference is taken, the definition of composite Simpson's rule appears.

$$\frac{4}{3}I_T(n) - \frac{1}{3}I_M(n/2) = \frac{b-a}{3n}[f(a) + f(b) + 4f(x_{even}) + 2f(x_{odd})]$$

3. $Q_3(f) = \frac{4h}{3}[2f(h) - f(2h) + 2f(3h)] \quad 0 \leq x \leq 4h$

(a) Given that this approximation uses 3 nodes, and therefore uses three function evaluations to approximate the integral, we can say that $Q_3(f)$ is of precision 3. This means $Q_3(f)$ can approximate integrals exactly for polynomials up to and including degree 3.

(b) We know that in general, the formula for the error in an odd degree Quadrature rule is

$$E_n = Kh^{n+1}p^n(\eta)$$

where h is the node spacing, p^n is a polynomial of degree $n+1$, and η is an unknown location on the interval. In order to find the value of K, we chose the function $f(x) = x^4$, where $f^{(iv)}(x) = 24$. It is known that

$$\int x^4 = \frac{x^5}{5}$$

so

$$\int_0^{4h} x^4 = \frac{(4h)^5}{5}$$

We also know the approximation, as it was given to be

$$\frac{4h}{3}[2f(h) - f(2h) + 2f(3h)]$$

If we expand each side as a Taylor series, we get

$$F(x) = \int f(x) = F(a) + hF'(a) + \frac{h^2}{2}F''(a) + \frac{h^3}{3!}F'''(a) + \ldots$$

where a is given as 0. We know that $F(0)$ is zero, since F is a one term polynomial. F can now be rewritten in terms of f, as the previously unknown term $F(a)$ is now gone.

$$F(x) = (x-a)f(a) + \frac{(x-a)^2}{2!}f''(a) + \frac{(x-a)^3}{3!}f'''(a) + \frac{(x-a)^4}{4!}f^{(iv)}(a)$$

What we are looking for is $F(4h)$. We use the Taylor expansion of F to find this value. As we know that $x - a = 4h$, we can make this substitution as well.

$$F(4h) = 4hf(a) + \frac{(4h)^2}{2!}f'(a) + \frac{(4h)^3}{3!}f''(a) + \frac{(4h)^4}{4!}f^{(iv)}(a)$$

Remembering that $F(a)$ was being approximated by $\frac{4h}{3}[2f(h) - f(2h) + 2f(3h)]$, we can expand each f term in the approximation to achieve

$$F^*(h) = \frac{4h}{3}[2f(a) +$$

$$[f(a) + 2hf'(a) + \frac{4h^2}{2!}f''(a) + \frac{8h^3}{3!}f'''(a) + \ldots] -$$

$$2[f(a) + 3hf'(a) + \frac{9h^2}{2!}f''(a) + \frac{27h^3}{3!}f'''(a) + \ldots]] + E$$

If we simply compare the $f^{(iv)}(x)$ terms in each of the Taylor expansions, we see that the term for F is $\frac{32h^5}{5!}f^{(iv)}(a)$, and the term for F^* is $\frac{4h}{3}[\frac{(2h)^5}{5!} - \frac{(3h)^5}{5!}]f^{(iv)}(a) = -\frac{211h^5}{5!}f^{(iv)}(a) + \ldots + E$

When we take the difference between the actual and the approximate terms, we finally see E

$$E = [\frac{32h^5}{5!} + \frac{211h^5}{5!}]f^{(iv)}(a) = \frac{241h^5}{5!} + \ldots$$

or, if we want to truncate the sequence by using an unknown a,

$$E = \frac{241h^5}{5!}f^{(iv)}(\eta)$$

4. Consider the value

$$A = \int_a^b f(x)dx$$

(a) We can let

$$A^* = \int_a^c f(x)dx$$

such that A^* satisfies

$$\frac{A^*}{A} = s$$

where $0 < s < 1$. We must first calculate the value of $\int_a^b f(x)dx$ numerically to within a specified tolerance. In order to do this, we can use any procedure we'd like. In this case, we will use the built in MATLAB routine `integral(fun,a,b)` to calculate an "exact" value of the area under $f(x)$ when $a \leq x \leq b$. This gives us a final result for the denominator. We also know the ratio these integrals should reach is s. There is only one unknown left, in the form of $\int_a^c f(x)dx$.

One way to approach solving for c is to use the same method to calculate $\int_a^c f(x)dx$ as was used to calculate $\int_a^b f(x)dx$. We must iterate through the domain $[a, b]$ in order to find a value of the integral that produces the correct ratio, at which point we are done.

We can start by setting $c = a$, and incrementing c by a node spacing h, which can begin as defined by the optimal node spacing for a Simpson's rule approximation, $h = (\frac{90}{24*10^6})^(1/5)$, where 10^{-6} is the desired error. We then iterate through the the domain, looking for values of c that satisfy the inequality $(s - error) \leq \frac{A^*}{A} \leq (s + error)$. In order to improve efficiency, when the ratio is exceeded, the iteration starts over at the most recent c such that $\frac{A^*}{A} \leq s$, but with the step size cut in half.

(b) Seen below is the implementation of the algorithm described in part (a).

Definition of $f(x)$ by `fun.m`

```
function y = fun(x)
y = x.^4;
```

`int_ratio.m` script to calculate c

```
function c = int_ratio(a,b,fun,s,tol)
% int_ratio.m
%   script to calculate a value c such that the area under a given curve
%       between x=a and x=c is s% of the area under the same curve from x=a to
%       x=b.
%   input:
%       a,b = interval of integration
%       @f = function to be integrated
%       s = ratio of areas
%       tol = tolerance of the result
%   output:
%       c = upper limit of integration to achieve desired ratio
```

```
clc;

% calculate integral using built in routine
A = integral(fun,a,b);
fprintf('Area under f(x) for 0 <= x <= 1: \n\t%.6f\n\n',A);

% calculate initial step size for c using error estimate for composite
% simpson's rule
step = ((90/10^6)/24)^(1/5);

% begin with starting guess cc. this value is updated every time the
% desired ratio is exceeded
cc = 0;

fprintf('c        \t\tN        \t\tN/A\n');
disp('--------------------------------------');

% iterate through c values with initial step size = step
% if no acceptable value is found, cut step size in half and repeat
% no more than 20 halvings of step is allowed

for iter = 1:20
    % iterate through possible c values, stopping when
    % |integral(@f,a,c)/integral(@f,a,b)| \in s=+-10^-6
    fprintf('Step size %.5f\n',step);
    accept = 0;
    for c = cc:step:b
        % evaluate integral
        N = integral(fun,a,c);

        %if mod((100000*c),2) == 0
            fprintf('%.6f\t',c);
            fprintf('%.6f\t',N);
            fprintf('%.6f\n',N/A);
        %end

        % save values of c that get you close to the ratio
        if abs(N/A) < s
            cc = c;
        end

        % see if ratio is satisfied within tolerance
        if (s-tol < abs(N/A)) && (abs(N/A) < s+tol)
            fprintf('\nAcceptable value of c has been found.\n\tc = %.6f\n',c);
           % set flag to show that a good c value was found
            accept = 1;
            return;
        end

        % stop once you've definitely exceeded the ratio
        if abs(N/A) > s+tol
```

```
            break
        end

    end

    % acceptable c has been found
    if accept == 1
        return;
    end
    % no c has been found, try again with smaller steps
    if accept == 0
        step = step / 2;
    end
    fprintf('\n');
end
```

Output of `int_ratio.m` script

```
c = int_ratio(0,2,@fun,0.5,0.000001)
Area under f(x) for 0 <= x <= 1:
        6.400000

c               N               N/A
------------------------------------------
Step size 0.08219
0.000000        0.000000        0.000000
0.082188        0.000001        0.000000
0.164375        0.000024        0.000004
0.246563        0.000182        0.000028
0.328750        0.000768        0.000120
0.410938        0.002344        0.000366
0.493126        0.005832        0.000911
0.575313        0.012605        0.001970
0.657501        0.024576        0.003840
0.739688        0.044287        0.006920
0.821876        0.075000        0.011719
0.904064        0.120788        0.018873
0.986251        0.186624        0.029160
1.068439        0.278470        0.043511
1.150626        0.403368        0.063026
1.232814        0.569531        0.088989
1.315001        0.786432        0.122880
1.397189        1.064893        0.166389
1.479377        1.417176        0.221434
1.561564        1.857074        0.290168
1.643752        2.400000        0.375000
1.725939        3.063076        0.478606
1.808127        3.865224        0.603941

Step size 0.04109
```

```
1.725939         3.063076          0.478606
1.767033         3.445510          0.538361

Step size 0.02055
1.725939         3.063076          0.478606
1.746486         3.249795          0.507780

Step size 0.01027
1.725939         3.063076          0.478606
1.736213         3.155330          0.493020
1.746486         3.249795          0.507780

Step size 0.00514
1.736213         3.155330          0.493020
1.741350         3.202284          0.500357

Step size 0.00257
1.736213         3.155330          0.493020
1.738781         3.178738          0.496678
1.741350         3.202284          0.500357

Step size 0.00128
1.738781         3.178738          0.496678
1.740065         3.190494          0.498515
1.741350         3.202284          0.500357

Step size 0.00064
1.740065         3.190494          0.498515
1.740708         3.196384          0.499435
1.741350         3.202284          0.500357

Step size 0.00032
1.740708         3.196384          0.499435
1.741029         3.199333          0.499896
1.741350         3.202284          0.500357

Step size 0.00016
1.741029         3.199333          0.499896
1.741189         3.200808          0.500126

Step size 0.00008
1.741029         3.199333          0.499896
1.741109         3.200071          0.500011

Step size 0.00004
1.741029         3.199333          0.499896
1.741069         3.199702          0.499953
1.741109         3.200071          0.500011

Step size 0.00002
1.741069         3.199702          0.499953
```

```
1.741089         3.199886         0.499982
1.741109         3.200071         0.500011

Step size 0.00001
1.741089         3.199886         0.499982
1.741099         3.199978         0.499997
1.741109         3.200071         0.500011

Step size 0.00001
1.741099         3.199978         0.499997
1.741104         3.200025         0.500004

Step size 0.00000
1.741099         3.199978         0.499997
1.741101         3.200001         0.500000

Acceptable value of c has been found.
         c = 1.741101
```